IN THE SPECIFICATION:

The specification as amended below with replacement paragraphs shows added text with underlining and deleted text with ~~strikethrough~~.

Please REPLACE paragraph [0007] on page 2 with the following amended paragraph:

[0007] Referring to FIGS. 1 and 2, a method of loading AV data into a memory and displaying the AV data will be described. A pickup device searches for a markup document STARTUP.HTM and loads the searched markup document STARTUP.HTM into a cache memory. Thereafter, the STARTUP.HTM is activated. At the same time, #1 AV data ~~①~~ selected by a user is loaded into a buffer memory and then displayed. Thereafter, #2 AV data ~~②~~ is loaded into the buffer memory and then displayed. Where a buffering of the #2 AV data ~~②~~ is complete, the pickup device jumps to a place on the interactive DVD where #3 AV data ~~③~~ is recorded and starts buffering the #3 AV data~~-③~~. At this time, the user may request a markup document ~~④~~ #4 A.HTM. In this case, the pickup device stops buffering the #3 AV data~~-③~~, searches for the markup document ~~④~~ #4 A.HTM, and loads the markup document ~~④~~ #4 A.HTM into the cache memory. While searching for the markup document ~~④~~ #4 A.HTM and loading it into the cache memory, the #3 AV data ~~③~~ is kept from being displayed. Therefore, the amount of data that can be buffered in the buffer memory is drastically decreased as the #3 AV data ~~③~~ still occupies the space in the buffer memory. Where the markup document ~~④~~ #4 A.HTM is activated, and the buffering of the #3 AV data ~~③~~ is complete, #5 AV data ~~⑤~~ is buffered. Thereafter, the pickup device jumps to a place where #6 AV data ~~⑥~~ is recorded. In the above method, all the data that has been buffered so far may disappear. In other words, where a reproduction of DVD-video images from a conventional interactive DVD in synchronization with markup documents is requested, for example, where a display of a specific actor's or actress's personal history whenever he or she appears on a screen is requested, the pickup device stops buffering AV data and begins searching for and caching the associated markup documents, and thus images may be discontinuously reproduced.

Please REPLACE paragraph [0020] on page 4 with the following amended paragraph:

[0020] To achieve the above and/or other aspect of the present invention, there is provided another apparatus for controlling a buffer which buffers a markup document to reproduce AV data in an interactive mode, comprising a buffer manager which manages the buffer to preload the markup document and outputs information of the buffer including buffering information of the markup document, wherein the buffering information includes information indicating that preloading of the markup document succeeded, information indicating that the preloading of the markup document failed, and information indicating that the preloading of the markup document is still ~~be~~ being conducted.

Please REPLACE paragraphs [0063] and [0064] on page 13 with the following amended paragraphs:

[0063] FIG. 8 illustrates a method of preloading files, the method corresponding to the operation 603 of FIG. 6. In operation 801, the presentation engine 5 identifies the path recorded in a link tag of the preload-list file and ~~draws~~ reads the preload-list file. In operation 802, the presentation engine 5 interprets the preload-list file, which includes a preload tag that has the paths and attributes of the files to be preloaded as parameters, and performs a preloading of the files.

[0064] FIG. 9A illustrates another method of preloading files, the method corresponding to the operation 603 of FIG. 6. In operation 901a, the presentation engine 5 interprets the API inserted into a body tag using parameters specifying the paths of the files to be preloaded and reads the files to be preloaded using the API. In operation ~~901b~~ 902a, the presentation engine 5 performs a preloading by interpreting the preload-list file that includes the paths and attributes of the files to be preloaded. Since the presentation engine 5 can determine the attributes of the files to be preloaded, it can process the files to be preloaded based on their attributes and store the processed files in a memory.

Please REPLACE paragraph [0068] on pages 14 and 15 with the following amended paragraph:

[0068] FIG. 12 illustrates the effects of a preloading process on an interactive DVD where AV data and HTML documents are recorded in the same manner as in FIG. 1. That is, FIG. 12 shows occupancy of the first memory 2 where MPEG-coded AV data is buffered and occupancy of the second memory 3 where a web resource is cached. Referring to FIGS. 1 and 12, the reader 1 searches for and reads STARTUP.HTM, and the presentation engine 5 interprets the preload information included in the STARTUP.HTM so that ~~④~~ #4 A.HTM is preloaded into the second memory 3. Where the STARTUP.HTM, which is loaded into the second memory 3, is activated, ~~①~~ #1 AV data is loaded into the first memory 2 and then displayed. Thereafter, ~~②~~ #2 AV data is loaded into the first memory 2 and then displayed. Where buffering of the ~~②~~ #2 AV data is completed, the reader 1 jumps to a place where ~~③~~ #3 AV data is recorded and starts buffering the ~~③~~ #3 AV data. At this time, where a user requests ~~④~~ #4 A.HTM, the presentation engine 5 reads ~~④~~ #4 A.HTM from the second memory 3 and displays the ~~④~~ #4 A.HTM. In this case, there is no need for the reader 1 to stop the buffering of the ~~③~~ #3 AV data, search the DVD 300 for the ~~④~~ #4 A.HTM, and then load the document ~~④~~ #4 A.HTM into the second memory 3. Therefore, the reader 1 can continue to buffer the ~~③~~ #3 AV data. Where the reader 1 completes the buffering of ~~⑤~~ #5 AV data and jumps to a place where ~~⑥~~ #6 AV data is recorded, the amount of data buffered in the first memory 2 may be reduced. However, the amount of data that has been buffered in the first memory 2 is sufficient so that a shortage in buffered data does not occur. In other words, even where there is a need to display DVD-video images, reproduced from an interactive DVD during the interactive mode, in synchronization with HTML documents, the reader 1 does not have to stop the buffering of AV data and then search for and cache the HTML documents. This is because the HTML documents have already been preloaded in the second memory 3. For example, synchronization display may be used where there is a need to display a specific actor's or actress's personal history whenever he or she appears on a screen.

Please REPLACE paragraph [0073] on pages 15 and 16 with the following amended paragraph:

[0073] The content decoder 52 may comprise an interpretation engine which parses and interprets the markup documents, and a browser which ~~draws~~ retrieves the markup documents from the interpretation engine and/or the network. Here, the markup documents correspond to various kinds of markup resources, ranging from markup text data written in HTML, CSS, or JAVASCRIPT to binary data, such as image data, audio data, or a Java program, which is referred to by markup documents. The markup documents are ~~drawn~~ retrieved from the disk 100 or the network by the buffer manager 51 in the ENAV engine 50.

Please REPLACE paragraph [0079] on page 17 with the following amended paragraph:

[0079] A release signal indicates that the predetermined data transferred from the ENAV buffer 30 to the content decoder 52, in response to the above-described reproduce signal, is no longer needed. For example, where a predetermined markup document is referred to five times in response to a reproduce signal, a release signal is generated five times. A counter value increases whenever a reproduce signal is generated and decreases whenever a release signal is generated. Where a counter value corresponding to a ~~predetermine~~ predetermined markup document reaches 0, i.e., where all reproduced markup documents are released, the released markup documents are deleted from the ENAV buffer 30 in response to a discard signal, which is described below.

Please REPLACE paragraphs [0085] and [0086] on pages 21 and 22 with the following amended paragraphs:

[0085] FIG. 16 illustrates a method of controlling a buffer state carried out by the content decoder 52 and the buffer manager 51, according to an embodiment of the present invention. In operation 1601, the content decoder 52 generates a fetch signal in response to a preload command. In operation 1602, the buffer manager 51 starts to read designated markup documents in response to the fetch signal. In operation 1603, the content decoder 52 determines, for example, whether all of the markup documents have been read and/or generates a report signal to determine, for example, a buffering state of the markup documents. In operation 1604, where an error occurs, the error is processed. In operation 1605, the buffer manager returns a signal indicating the buffering state of the markup documents in response to the report signal. In operation 1606, the content decoder 52 generates a ~~retrieves~~ retrieve signal to use the markup documents. In operation 1607, the buffer manager 51 ~~transfers~~ transmits the designated markup documents to the content decoder 52 in response to the retrieve signal. In operation 1608, the content decoder 52 ~~presented~~ presents the designated markup documents. In operation 1609, where the designated markup documents are no longer to be used, the content decoder 52 generates a release signal. In operation 1610, the buffer manager 51 decreases a current value of a counter by 1 to indicate, for example, use of a corresponding one/ones of the designated markup documents. In operation 1611, the content decoder 52 generates a discard signal to delete the designated markup documents. In operation 1612, the buffer manager 51 deletes the designated markup documents from the ENAV buffer 30 in response to the discard signal.

[0086] FIG. 17 illustrates a data storage medium, for example, a disk, on which AV data and markup documents are recorded, according to an embodiment of the present invention. The data storage medium further comprises control information which is used to identify buffering state information of a markup document to be preloaded. For example, a startup document STARTUP.HTM includes a preload list file STARTUP.PLD to seamlessly reproduce ~~files~~ AV data while displaying markup documents ranging from A.HTM to D.HTM.

Please REPLACE paragraphs [0090] and [0091] on pages 22 and 23 (including the list of computer code on page 23) with the following amended paragraphs:

[0090] FIG. 20 illustrates a predetermined order in which the markup documents and the AV data recorded on the disk of FIG. 17 are reproduced. For example, where each scene begins, whether reference files of a preload list file corresponding to the scene have been read is checked using an ~~IsCashed~~ IsCached API. Where reading of the reference files has been successfully completed, ~~HTM~~ HTML documents are read and reproduced. Thereafter, markup documents that have already been reproduced are discarded using a Discard API.

[0091] For example, to seamlessly reproduce AV data while displaying markup documents ranging from ~~STARTUP.HTM,~~ A.HTM,~~ and~~ to D.HTM, the preload list file STARTUP.PLD included in the startup document STARTUP.HTM is specified as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PRELOAD PUBLIC "-//DVD//DTD DVD Preload List 1.0//EN"
"http://www.dvdforum.org/enav/dvd-preload-list.dtd"-->
<filedef type="text/xml" src="disc://dvd_enav//a.htm" />
<filedef type="text/xml" src="disc://dvd_enav//a.pld" />
<filedef type="image/png" src="dvd://dvd_enav//a1.png" />
<filedef type="image/png" src="dvd://dvd_enav//a2.png" />
<filedef type="image/png" src="dvd://dvd_enav//a3.png" />
<filedef type="text/xml" src="disc://dvd_enav//b.htm" />
<filedef type="text/xml" src="disc://dvd_enav//b.pld" />
<filedef type="audio/au" src="dvd://dvd_enav//b1.au" />
<filedef type="image/png" src="dvd://dvd_enav//b2.png" />
<filedef type="image/png" src="dvd://dvd_enav//b3.jpg" />
<filedef type="text/xml" src="disc://dvd_enav//c.htm" />
<filedef type="text/xml" src="disc://dvd_enav//c.pld" />
<filedef type="image/png" src="dvd://dvd_enav//c1.png" />
<filedef type="image/png" src="dvd://dvd_enav//c2.png" />
<filedef type="image/png" src="dvd://dvd_enav//c3.png" />
<filedef type="text/xml" src="disc://dvd_enav//d.htm" />
<filedef type="text/xml" src="disc://dvd_enav//d.pld" />
<filedef type="image/png" src="dvd://dvd_enav//d1.png" />
<filedef type="image/png" src="dvd://dvd_enav//d2.png" />
</preload>
```

Please REPLACE paragraph [0095] on page 26 with the following amended paragraph:

[0095] It is understood that a system which uses the present invention also includes permanent or removable storage, such as magnetic and optical discs, RAM, ROM, ~~a carrier wave medium,~~ etc., on which the process and data structures of the present invention can be stored and distributed. The ~~operations~~ process and data structures of the present invention can also be distributed via, for example, downloading over a network such as the Internet, or transmitted via a carrier wave.